ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY

Affiliated to M. G. University, Kottayam and approved by AICTE, ASIET is the first self – financing technical education centre to be awarded the ISO 9001: 2008 certification. It was established in 2001with the aim of providing value added technical education with a flair of professional excellence and ethical values.

Automation System for ac using raspberry pi  
  
Software Requirements Specification

© 2014 ASIET

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 21/03/16 | 1.0 | Initial Version | Aravind prakash |
|  |  |  | Arjun anilkumar |
|  |  |  | mihir |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Purpose 2

1.1. Scope 2

1.2. Definitions, Acronyms, Abbreviations 2

1.2.1. Definitions 2

1.2.2. Acronyms 2

1.3. References 2

1.4. Overview 3

2. Overall Description 4

2.1. Product Perspective 4

2.2. Product Architecture 4

2.3. Product Functionality/Features 4

2.4. User Characteristics 5

2.5. Constraints 5

2.6. Assumptions and Dependencies 5

3. Specific Requirements 6

3.1. Functional Requirements 6

3.2. External Interface Requirements 6

3.3. Internal Interface Requirements 6

3.4. Internal Data Requirements 6

3.5. Design and Implementation Constraints 6

4. Non-Functional Requirements 7

4.1. Safety Requirements 7

4.2. Security and Privacy Requirements 7

4.3. Environmental Requirements 7

4.4. Computer Resource Requirements 7

4.4.1. Computer Hardware Requirements 7

4.4.2. Computer Hardware Resource Utilization Requirements 7

4.4.3. Computer Software Requirements 8

4.4.4. Computer Communication Requirements 8

4.5. Software Quality Factors 8

4.6. Packaging Requirements 8

4.7. Precedence and Criticality of Requirements 8

5. Qualification Provisions 9

# Purpose

This specification document explains about a Raspberry Pi based ac Automation System where an end user can control device(s) remotely from anywhere in the world. The end user can authenticate into a Web Interface and one will be provided with a dashboard from which she/he can control devices as well as acknowledge the present status of connected devices so as to control the on and off and the temperature control.

## Scope

The system being developed is an Automation System for ac where one can control the device remotely from anywhere in the world. A user can log on to a android Application through which she/he can switch ON or switch OFF connected devices. They can also see the status of presently connected devices too. A user can also access the Application from handheld devices such as a Mobile Phone with Internet connectivity and then supervise the connected devices.

## Definitions, Acronyms, Abbreviations

### Definitions

USB – Universal Serial Bus (USB) is an industry standard developed in the mid-1990s that defines the cables, connectors and communications protocols used in a bus for connection, communication, and power supply between computers and electronic devices.

Raspberry Pi-The Raspberry Pi is a low cost, **credit-card sized computer** that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It’s capable of doing everything you’d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

### Acronyms

RPi – Raspberry Pi

HDMI – High Definition Multimedia Interface

GPU – Graphics Processing Unit

GPIO – General Purpose Input Output

BaaS – Backend-as-a-Service

## References

[I] AI-Ali A. R. and AI-Rousan M., "Java-based home automation system", IEEE Transactions on Consumer Electronics, vol. 50, no. 2, pp. 498- 504, 2004.

[2] Ali M., V laskamp J.H.A, Eddiny N.N. , Falconer B. and Oram c., "Technical Development and Socioeconomic Implications of the as a Learning Tool in Developing Countries", 5th Computer Science and Electronic Engineering Conference (CEEC), pp. 103- 108, 2013.

## Overview

Existing System

*Costly Hardware*: Most of the vendors provide a wired system which is accomplished by setting up a new wiring and this adds to the overall implementation cost. They also include costly hardware controllers such as touch screen panels which contribute to the increased cost.

*Non-energy Efficient Systems*: The components or the peripherals that make up the Home Automation system often are quite energy consuming by themselves. This poses serious environmental concerns as well adds to additional expense to be bound by the customer or the end user.

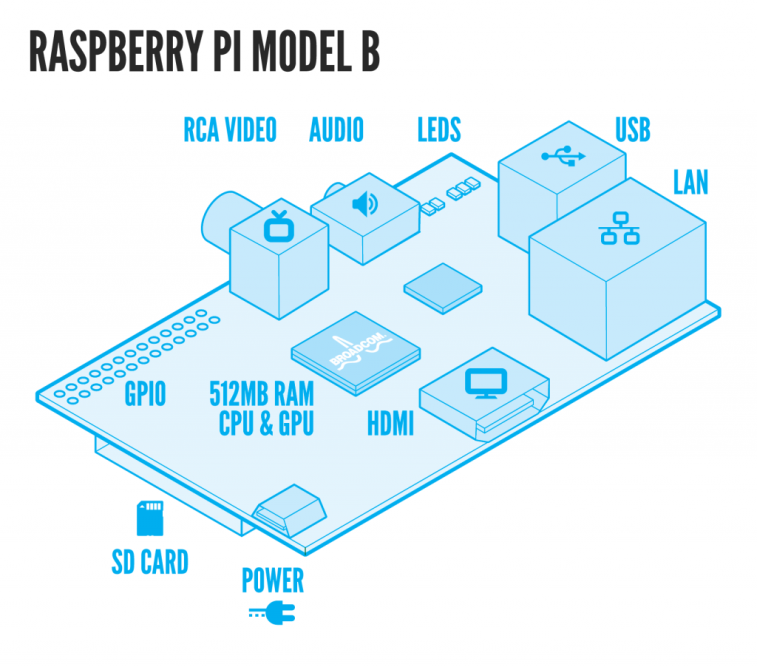
Another shortcoming is that it requires the end user to be in visibility of the device being controlled. In other words a remote control based system where the user has to be in the proximity of the sensor actuating the devices.

Proposed System

**About the Raspberry Pi Board:**

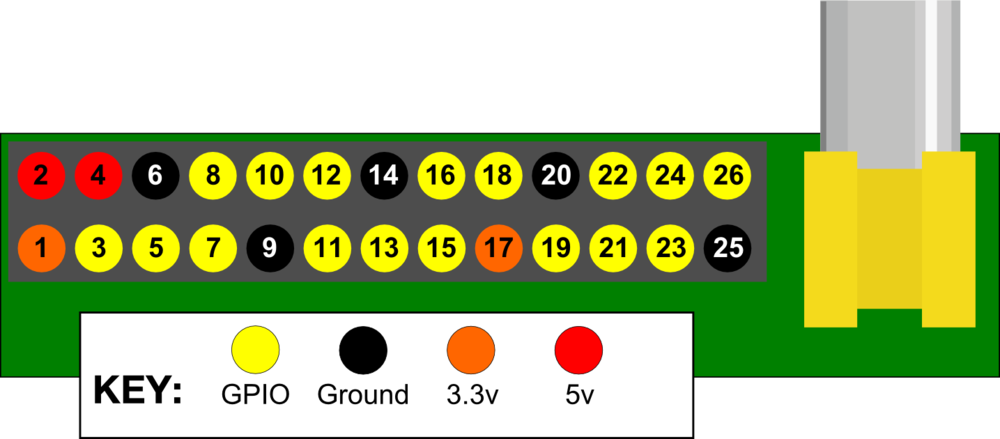
In the proposed system, we will use an RPi as the vital controlling unit. The RPi isn’t just a simple µ-controller rather it is a credit-card sized single board computer with a powerful µ-processor. It provides all basic functions such as processing power, Internet connectivity, video/audio playback, storage access via an externally mounted SD card and ports such as USB, Ethernet, Display, Camera and HDMI. Some of the advantages of an RPi over some other boards available in the market include:

* *Less Cost*. The cost of an RPi is just 25 USD which is far less compared to other equivalent boards like Intel Galileo, BeagleBone, HummingBoard etc.
* *Power Consumption*. The power needed for the working of an RPi is close to 3W.



At the core of the RPi, is its 700 MHz processor manufactured by Broadcom. The RPi uses a SoC. This SoC has everything been integrated to a single chip which includes the processing unit, 512 MB RAM and the VideoCore IV GPU. The CPU is built on RISC Architecture.

One powerful feature of the RPi is the row of **General Purpose Input Output (“GPIO”)** pins.



The GPIO pins form an essential part in controlling the real world objects. The other major components of the proposed Home Automation system includes Internet connection (Ethernet or Wi-Fi), Device under control, Remote Controlled Outlets, Jumper wires and breadboard.

**PORTS AND THEIR USAGE**

* VIDEO (Camera Module): For Video Access. Video out and Video in ports are available. There are three command line applications provided for stills, video, and stills output uncompressed. These applications provide the typical features like Set image size, compression quality, exposure mode, ISO.
* Audio: There is a standard 3.5mm jack for audio out to an amplifier. You can add any supported USB microphone for audio in, or using the I2S interface you can add a codec for additional audio I/O.
* LAN: Raspberry Pi LAN chip is smart enough to reconfigure itself for direct network connections.
* HDMI port: for coding in the Raspberry Pi a monitor is connected to the HDMI port. We may use a television monitor.

# Overall Description

## Product Perspective

The system as a whole allows end users to control ac remotely. S/he can authenticate oneself to an android app and then control the various devices remotely as well as acknowledge the statuses of the currently active systems. The Web Application can also be accessed from a hand-hand held device thereby one can control the system even on the go.

Apart from this it also is an economical and environment friendly approach to manipulate devices or control systems where in direct human intervention is impossible.

## Product Architecture

The below diagram depicts the overall architecture of the system:

User logs onto

The Web App

Raspberry Pi

Android app

USER

Remote Outlet from raspberry pi

Control ac

ON/OFF[mode 1]

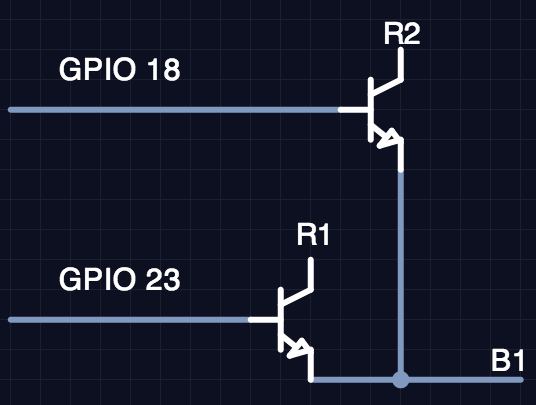
Temp control[mode 2]

**The end user is provided with an android app that runs on any android phone that connects the user t0 the Raspberry Pi**. The user can login to the android App with a set of pre-defined credentials.

The Dashboard allows the User to know the status of the currently connected device(s). It also provides buttons to toggle the state (ON/OFF) of the connected device. When the user turns a device ON by toggling the button, a request will be send to the server that hits RPi to activate respective pin.

On receiving this request, RPi activates the corresponding GPIO pin. Activating the GPIO pin, will in turn toggle the switch on the Remote Control there by actuating the AC Outlet for which the switch was responsible for. This will trigger to the device connected to that AC Outlet.

The following diagram shows the ground connection between a switch on the Remote Control and the GPIO pin of the RPi:



The user can also access videos’s and sound recordings with help of the camera module (video input/output) and audio input/output ports available on the board.

## Product Functionality/Features

Some the significant features of the proprosed Home Automation system include:

* *Cost Effective*. The RPi used as the central controlling system is very cheap.
* *Energy Efficient*. The total power required for the operation of whole system including the operational power of the Remote Control and the RPi will not exceed 5W. Thus the end user is free from energy concerns!
* *Easily Extensible*. Since there are around 17 GPIO pins available on the RPi, you can control up to 17 devices provided that we have the requisite amount of AC Outlets. Also, the maximum number of GPIOs can theoretically be indefinitely expanded by making use of the i2c or SPI bus.
* *Automate From Anywhere*. Since the chief controlling platform is an android based application, it can be accessed from anywhere provided that you have an Internet connection. The app can also be accessed via hand-held devices such a Tablet or Mobile Phone thus enabling the end user to control the devices even on the move!
* *Acknowledgement*. Whenever a particular function could not be carried out, immediately an acknowledgement will be provided to the User.
* To implement in college, the device to be controlled connected to the remote controller socket which in turn is connected to the RPi. The device can switched ON/OFF using the Web Application.

## User Characteristics

The end user can control the whole system via the android Application . She/he can login to the system with his or her credentials. Some of the basic functions that a user can perform include:

* *Switch a Device ON or OFF*. Every user logging into the system is provided with a Dashboard which shows the number of systems currently connected to the RPi. Each device shown on the Dashboard will have button (representing a switch) below it which can be used to toggle the state of the device.
* *Device Status*. The dashboard will also show the status of each device as to whether it is ON or OFF. Thus a user can perceive if the device is ON or OFF before taking an action.
* *Device Activity History*. A User can also view the details such as how long a device was active for the day.
* *Temperature control*. user con set the temperature limits to control temperature of the room automatically.

## Constraints

One of the main constraints is:

* API latencies in carrying out database transactions.
* *Response Time*. The device controlled by the RPi responds in less than 5 seconds. This means that the round trip delay starting from turning the device on via the android App and the App sending the request to the server which connects it to the pi so as to control the GPIO pins and the Pins activating the Remote Controlled Outlet altogether should be less than 5 seconds.

## Assumptions and Dependencies

It is assumed that the device controlled by the GPIO pin is functional. The only intimation that the end user would know is the status of the pin: active/in-active which signifies that there is a supply of power into the device. There isn’t means as of now to determine if the device or working or not.

In case of an Internet failure the system provides a message to user.

# Specific Requirements

## Functional Requirements

The different functional units required for the operation of the system are listed as below:

* *User Authentication*. The credentials provided by the user will be equated against the database hosted on a cloud based BaaS such as Firebase. Only if validated will s/he be allowed to the Dashboard.
* *Controlling Devices*. There will be dashboard allows the User to switch ON and OFF various devices. It also lets the User know the status of device.
* *Temperature control*.the app is set to work in two modes.the second mode allows the user to set the temperature limits so as to control the temperature automatically.
* *Error Handling*. The android Application will also house an Error Handling module to respond at times when a request couldn’t be completed due to the malfunction of the GPIO pins.
* *Web Server*. An external framework named Flask will be used to setup a Web Server on the RPi.
* *android Application*. An android based Application will be hosted on any android phone which provides facilities such as login and interactive dashboard in order to control the connected devices.

## External Interface Requirements

The external interfaces required as enumerated as below:

* 10/100 Mbit/s Ethernet
* Internet Connection (Ethernet or Wi-Fi)
* USB bandwidth 480Mbits/s
* Remote Controlled Outlets
* 6 2N2222A Transistors
* Jumper wires and breadboard
* Raspberry Pi

## Internal Interface Requirements

The internal interface required includes:

1 - internal relay system

2 – wi fi module or Ethernet set-up

## Design and Implementation Constraints

One of the main constraints is:

* *Lack of Storage Space/Processing Power*. Since the RPi has very limited processing power and there is hardly any disk space as it solely depends on the mounted SD card, we cannot setup an RDBMS on the RPi to manage the user accounts of the android App.
* *Response Time*. The device controlled by the RPi responds in less than 5 seconds. This means that the round trip delay starting from turning the device on via the android App and the App sending the request to the GPIO pins via server and the Pins activating the Remote Controlled Outlet altogether should be less than 5 seconds.

# Non-Functional Requirements

## Safety Requirements

The improper functioning of AC outlets could be potential haphazard. Thus one must ensure that proper connectivity of the outlet and the device has been put to place.

## Security and Privacy Requirements

The user credentials should be kept confidential. The lack of which could lead to opening up access to wrong hands.

## Environmental Requirements

The environmental requirements include:

* *Raspbian*. The RPi operates only a Raspbian Operating System.
* *Android studio for the development of android app*.

## Computer Resource Requirements

### Computer Hardware Requirements

The computer hardware requirements are as below:

* Raspberry Pi
* Internet Connection (Ethernet or Wi-Fi)

\* - *Only for initial configuration of the Raspberry Pi*

### Computer Hardware Resource Utilization Requirements

The following are the upper limit on the resource utilization map:

* *Storage Space* - SD card 8GB
* *Processing Power* - 512 MB
* *Processing Speed* - 700Mhz

### Computer Software Requirements

Given below are the software requirements:

* *Operating System* - Raspbian OS
* *Web Server*
* *Compiler* – Python/c

## Software Quality Factors

The various quality factors that uniquely distinguish the proposed system include:

* *Energy Efficient*. Since the whole system will not consume more 5W of power, it makes very energy efficient thus making the end user from any concerns.
* *Cost Effective*. The software components are purely open source and thus we aren’t bound to any licensing costs. Moreover the RPi is very cheap yet very efficient too.
* *Ease of Use*. In order to switch ON or OFF a device, she/he can simply log on to the Web App and control the same from the interactive dashboard.
* *Scalability*. Since there are17 GPIO pins available on the RPi board, we can control up to 17 Devices. Also, with the help of I2C, one can actually extend the number of GPIO pins indefinitely. This clearly shows that we can scale in the number of connected devices with ease.

## Packaging Requirements

In order to make a re-distributable package, the following are the components that need to be taken care of:

* *Operating System for the RPi* – Raspbian [Kernel Version: 3.12]
* *Web Server* – Flask v0.10.1
* *Circuit Components* –
  + 6 2N2222A Transistors
  + Remote Controller AC Outlets

# Qualification Provisions

1. *Test Case*: **Logging into the Web App**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sl No | Module | Test Case | User | Expected I/P | Expected O/P | Remark |
| 1 | Login | User Login | User | Attempt to authenticate with a valid Username and Password | Re-directed to the Dashboard | Success |
| 2 | Login | User Login | User | Attempt to authenticate with an invalid Username and Password | Re-direct to the Error Page | Failure |

1. *Test* *Case*: **Switching ON/OFF a Device**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sl No | Module | Test Case | User | Expected I/P | Expected O/P | Remark |
| 1 | DeviceON | Switch ON Device | User | Toggle the button below the device icon to switch it ON | Connected device begins to work | Success |
| 2 | DeviceON | Switch ON Device | User | Toggle the button below the device icon to switch it ON | Device does not work; Provides an error message | Failure |
|  | DeviceOFF | Switch OFF Device | User | Toggle the button below the device icon to switch it OFF | Connected device stops working | Success |

1. *Test Case*: **Viewing Device Activity History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sl No | Module | Test Case | User | Expected I/P | Expected O/P | Remark |
| 1 | View Details | View Activity History | User | Click on the Device Icon | Number of Active Hours | Success |

1. *Test Case*: **mode 2[temperature control]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sl No | Module | Test Case | User | Expected I/P | Expected O/P | Remark |
| 1 | Temperature settin[limits] | Set the limits | User | Click Access current temperature limit | Set the limits[upper and lower] | Success |
| 2 | Temperature setting[different tem] | Set the limits | User | Click Access current temperatue limit | Not able to set or change | Failure |